



CERTIFIED COPY OF
PRIORITY DOCUMENT



J11046 U.S. PTO

09/851405



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto
is a true copy from the records of the Korean Industrial
Property Office.

00153170

출원번호 : 특허출원 2000년 제 24608 호
Application Number

출원년월일 : 2000년 05월 09일
Date of Application

출원인 : 삼성에스디에스 주식회사
Applicant(s)



2001 년 02 월 05 일

특

허

청

COMMISSIONER



J1046 U.S. PTO
09/851405
05/08/01

KOREAN INDUSTRIAL PROPERTY OFFICE

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Industrial Property Office.

Application Number: Patent Application No. 00-24608

Date of Application: 9 May 2000

Applicant(s): Samsung SDS Co., Ltd.

5 February 2001

COMMISSIONER

[Document] Application
[Right] Patent
[Receiver] Commissioner
[Document No.] 0001
[Filing Date] 9 May 2000
[IPC] H04L
[Title] Software test system and method

[Applicant]
[Name] Samsung SDS Co., Ltd.
[Applicant code] 1-1998-001790-3

[Attorney]
[Name] Young-pil Lee
[Attorney code] 9-1998-000334-6
[General Power of Attorney
Registration No.] 1999-050326-4

[Attorney]
[Name] Hyuk-keun Cho
[Attorney code] 9-1998-000544-0
[General Power of Attorney
Registration No.] 2000-224538-0

[Attorney]
[Name] Hae-young Lee
[Attorney code] 9-1999-000227-4
[General Power of Attorney
Registration No.] 2000-004535-8

[Inventor]

[Name] Young-seok Lee
[Resident
Registration No.] 650315-1052030
[Zip code] 136-060
[Address] 101-1408, Samsung Apt., Donam-dong
Seongbuk-gu, Seoul, Rep. of Korea
[Nationality] Rep. of Korea

[Request for Examination] Yes



1020000024608

2001/2/5

[Purpose] We file the present application according to Article 42 of the Patent Law
also Request for Examination according to Article 60 of the Patent Law.
Attorney Young-pil Lee
Attorney Hyuk-keun Cho
Attorney Hae-young Lee

[Fee]
[Basic fee] 17 Sheet(s) 29,000 won
[Additional fee] 0 Sheet(s) 0 won
[Priority claiming fee] 0 Case(s) 0 won
[Examination fee] 3 Claim(s) 205,000 won
[Total] 234,000 won

[Enclosures]
1. Abstract and Specification (and Drawings) 1 original

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0001
【제출일자】	2000.05.09
【국제특허분류】	H04L
【발명의 명칭】	소프트웨어 테스트 시스템 및 그 방법
【발명의 영문명칭】	Software test system and method
【출원인】	
【명칭】	삼성에스디에스 주식회사
【출원인코드】	1-1998-001790-3
【대리인】	
【성명】	이영필
【대리인코드】	9-1998-000334-6
【포괄위임등록번호】	1999-025433-5
【대리인】	
【성명】	조혁근
【대리인코드】	9-1998-000544-0
【포괄위임등록번호】	2000-002836-5
【대리인】	
【성명】	이해영
【대리인코드】	9-1999-000227-4
【포괄위임등록번호】	2000-002829-9
【발명자】	
【성명의 국문표기】	이영석
【성명의 영문표기】	LEE, Young Seok
【주민등록번호】	650315-1052030
【우편번호】	136-060
【주소】	서울특별시 성북구 돈암동 삼성아파트 101동 1408호
【국적】	KR
【심사청구】	청구

【취지】

특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인

이영필 (인) 대리인

조혁근 (인) 대리인

이해영 (인)

【수수료】

【기본출원료】 17 면 29,000 원

【가산출원료】 0 면 0 원

【우선권주장료】 0 건 0 원

【심사청구료】 3 항 205,000 원

【합계】 234,000 원

【첨부서류】

1. 요약서·명세서(도면)_1통

【요약서】**【요약】**

소프트웨어 테스트 시스템 및 그 방법이 개시된다. 컴퓨터상에서 실행되는 소프트웨어를 테스트하는 본 발명에 따른 소프트웨어 테스트 시스템은 소프트웨어의 오브젝트들을 실행하기 위한 명령어들을 함수화하여 기록한 함수 라이브러리 파일, 소프트웨어의 각 오브젝트를 나타내는 오브젝트 인식값을 소프트웨어를 테스트하고자 하는 순서에 따라 순차적으로 기록하며, 각 오브젝트 인식값들은 오브젝트 구분자에 의해 구분되는 오브젝트 파일 및 오브젝트 파일로부터 오브젝트 인식값을 순차적으로 읽어들이 실행하고자 하는 오브젝트를 인식하고, 인식된 오브젝트를 실행하기 위한 함수를 함수 라이브러리 파일로부터 호출하여 실행하는 실행 프로그램을 포함을 특징으로 하며, 타겟 소프트웨어의 변경에 따른 유지 및 보수가 매우 간단히 이루어지며, 다른 소프트웨어 및 제품 테스트에 적용시 함수 라이브러리 파일에서 일부 함수를 재정의하는 것만으로도 매우 쉽게 적용할 수 있다.

【대표도】

도 2

【명세서】**【발명의 명칭】**

소프트웨어 테스트 시스템 및 그 방법{Software test system and method}

【도면의 간단한 설명】

도 1은 종래에, 테스트 시나리오대로 타겟 소프트웨어를 실행할 때 생성되는 스크립트 파일들을 나타낸다.

도 2는 본 발명에 따른 소프트웨어 테스트 시스템의 개념을 나타내는 도면이다.

도 3a 내지 도 3c는 도 2에 도시된 오브젝트 파일(14), 실행 프로그램(12) 및 함수 라이브러리 파일(10)의 일 예를 각각 나타내는 도면이다.

도 4는 본 발명에 따른 소프트웨어 테스트 방법의 일 실시예를 설명하기 위한 플로우차트이다.

【발명의 상세한 설명】**【발명의 목적】****【발명이 속하는 기술분야 및 그 분야의 종래기술】**

<5> 본 발명은 소프트웨어 테스트에 관한 것으로, 특히, 유지 및 보수가 간편한 소프트웨어 테스트 시스템 및 그 방법에 관한 것이다.

<6> 일반적으로, 테스트하고자 하는 타겟(target) 소프트웨어를 테스트하기 위한 테스트 툴은 많은 량의 스크립트(script) 파일들을 생성하게된다. 특히, 미국 레이셔널(Rational)사의 팀테스트(TeamTest)와 같이, 선형 스크립트 기술을 이용하는 테스트 툴의 경우, 스크립트 파일은 테스트 케이스 수에 비례하여 생성된다. 예컨대, 팀테스트를

이용하여 타깃 소프트웨어를 테스트하고자하는 경우, 사용자는 먼저 테스트 시나리오를 만들고, 테스트 툴을 가동한 상태에서 고안된 테스트 시나리오대로 타깃 소프트웨어를 한 번씩 실행한다. 팀테스트는 테스트 시나리오대로 타깃 소프트웨어를 실행할 때마다 해당 스크립트 파일들을 생성하며, 생성된 스크립트 파일들에 의해 타깃 소프트웨어를 자동적으로 실행하면서 타깃 소프트웨어를 테스트한다.

<7> 도 1은 테스트 시나리오대로 타깃 소프트웨어를 실행할 때 생성되는 스크립트 파일들을 나타내며, 이 때 테스트 케이스 수는 n 이다. 도 1을 참조하면, 테스트 케이스수 n 만큼 스크립트 파일들이 생성되며, 각 스크립트 파일에는 타깃 소프트웨어를 실행하기 위한 명령어(예컨대, 'Window SetContext', 'MenuSelect', 'PushButton')와 데이터('제목없음 - 메모장', '열기', '취소')가 혼재되어 있다.

<8> 한편, 이상에서와 같이 선형 스크립트 기술을 이용한 소프트웨어 테스트는 타깃 소프트웨어가 변경될 경우, 해당 스크립트 파일을 수정하거나 부분적으로 재실행하여 스크립트 파일을 다시 생성해야 한다. 만약, 해당 스크립트 파일을 수정하기 힘들거나 부분적으로 새로운 스크립트 파일의 재생성이 불가능하다면, 스크립트들을 모두 폐기하고 테스트 시나리오에 따른 스크립트 파일들을 처음부터 다시 생성해야 한다. 이처럼, 기존의 스크립트 파일들을 폐기하고 새로운 스크립트 파일들을 생성하 데 드는 시간은 다음 수학적 식 1에 의해 측정될 수 있다.

<9> 【수학적 식 1】

$$\text{스크립트 재생성시간} = \text{타깃소프트웨어 변경율} \times \text{스크립트를 만드는데 드는 평균시간}$$

<10> \times 전체 스크립트수 \times 예수(R)

<11> 여기서, 예수(R)는 프로젝트 규모, 스크립트 사이즈, 전체 스크립트수, 기능수등을 고려하여 부여하여야 하나, 여기서는 1로 정한다. 그리고, 전체 스크립트수를 50개로 한정하고, 타깃 소프트웨어의 변경 화면율을 5%라 하고, 단위 스크립트를 만드는데 드는 시간이 0.3시간이라 할 때, 스크립트 재생성 시간은 0.75시간이 된다. 만약, 타깃 소프트웨어의 변경 규모는 작으나 전체 화면의 50%에 걸쳐 변경이 발생된다면, 즉, 타깃 소프트웨어의 변경 화면율이 50%라면 스크립트 재생성시간은 7.5시간이 된다. 결국, 타깃 소프트웨어의 변경에 따른 소프트웨어 테스트 툴의 유지 및 보수가 복잡하고 많은 시간이 소요됨을 알 수 있다.

<12> 이상에서와 같이, 소프트웨어를 테스트하기 위해 일반적으로 쓰이는 선형 스크립트 기술은 타깃 소프트웨어에 약간의 변경만 발생되어도 스크립트 파일들중 해당 스크립트를 수정하거나 재생성해야 하며, 최악의 경우 전체 스크립트를 다시 생성해야 한다. 또한, 타깃 소프트웨어의 테스트 케이스수에 비례하여 스크립트 파일들이 생성되므로 테스트 케이스가 많을수록 스크립트 파일의 유지 및 보수는 더욱 어려워진다.

<13> 결국, 종래의 선형 스크립트 기술을 이용하여 타깃 소프트웨어를 테스트하면, 테스트 케이스가 많을수록, 타깃 소프트웨어 프로그램 변경이 잦을수록, 소프트웨어 테스트에 걸리는 시간과 노력이 많이 들게된다.

【발명이 이루고자 하는 기술적 과제】

<14> 본 발명이 이루고자 하는 기술적 과제는 테스트 대상 소프트웨어의 변경에 따른 유지 보수가 간단한 소프트웨어 테스트 시스템을 제공하는 데 있다.

<15> 본 발명이 이루고자 하는 다른 기술적 과제는 상기 소프트웨어 테스트 시스템에서 수행되는 테스트 방법을 제공하는 데 있다.

<16> 본 발명이 이루고자 하는 또 다른 기술적 과제는 상기 소프트웨어 테스트 방법을 컴퓨터에서 실행 가능한 프로그램 코드로 기록한 기록 매체를 제공하는 데 있다.

【발명의 구성 및 작용】

<17> 상기 과제를 이루기 위해, 컴퓨터상에서 실행되는 소프트웨어를 테스트하는 본 발명에 따른 소프트웨어 테스트 시스템은 소프트웨어의 오브젝트들을 실행하기 위한 명령어들을 함수화하여 기록한 함수 라이브러리 파일, 소프트웨어의 각 오브젝트를 나타내는 오브젝트 인식값을 소프트웨어를 테스트하고자 하는 순서에 따라 순차적으로 기록하며, 각 오브젝트 인식값들은 오브젝트 구분자에 의해 구분되는 오브젝트 파일 및 오브젝트 파일로부터 오브젝트 인식값을 순차적으로 읽어들이어 실행하고자 하는 오브젝트를 인식하고, 인식된 오브젝트를 실행하기 위한 함수를 함수 라이브러리 파일로부터 호출하여 실행하는 실행 프로그램을 포함한다.

<18> 상기 다른 과제를 이루기 위해, 컴퓨터상에서 실행되며, 테스트하고자 하는 타겟 소프트웨어를 실행하는 명령어를 함수로 일반화시킨 함수 라이브러리 파일을 구비하는 소프트웨어 테스트 시스템에서, 타겟 소프트웨어를 테스트하는 본 발명에 따른 소프트웨어 테스트 방법은 (a)타겟 소프트웨어의 각 오브젝트를 나타내는 오브젝트 인식값을 소프트웨어를 테스트하고자 하는 순서에 따라 순차적으로 기록하며, 각 오브젝트 인식값들이 오브젝트 구분자에 의해 구분되는 오브젝트 파일을 생성하는 단계, (b)오브젝트 파일에 기록된 오브젝트 인식값들을 하나씩 순차적으로 읽어들이고, 읽어들이는 오브젝트 인식값에 해당하는 오브젝트를 실행하기 위한 함수를 함수 라이브러리 파일로부터 호출하는

단계, (c)(b)단계에서 호출된 함수의 실행에 필요한 소정 개수의 함수 인자들로서 (a)단계에서 읽어들이는 오브젝트 인식값 다음에 연속하는 오브젝트 인식값들을 읽어들이며, (b)단계에서 호출된 함수를 실행하는 단계 및 (d)오브젝트 파일에 실행되지 않은 오브젝트 인식값이 있으면 (b)단계로 진행하여 테스트를 계속 진행하고, 없으면 테스트를 종료하는 단계로 이루어진다.

<19> 상기 또 다른 과제를 이루기 위해, 컴퓨터상에서 실행되며, 테스트하고자 하는 타깃 소프트웨어의 명령어를 함수로 일반화시킨 함수 라이브러리 파일 및 타깃 소프트웨어의 각 오브젝트를 나타내는 오브젝트 인식값을 소프트웨어를 테스트하고자 하는 순서에 따라 순차적으로 기록하며, 각 오브젝트 인식값들은 오브젝트 구분자에 의해 구분되는 오브젝트 파일을 구비하는 소프트웨어 테스트 시스템에서 타깃 소프트웨어를 테스트하는 본 발명에 따른 프로그램을 기록한 기록 매체는 (a)오브젝트 파일에 기록된 오브젝트 인식값들을 하나씩 순차적으로 읽어들이고, 읽어들이는 오브젝트 인식값에 해당하는 오브젝트를 실행하기 위한 함수를 함수 라이브러리 파일로부터 호출하는 단계, (b)(a)단계에서 호출된 함수의 실행에 필요한 소정 개수의 함수 인자들로서 (a)단계에서 읽어들이는 오브젝트 인식값 다음에 연속하는 오브젝트 인식값들을 읽어들이며, (a)단계에서 호출된 함수를 실행하는 단계 및 (c)오브젝트 파일에 실행되지 않은 오브젝트 인식값이 있으면 (a)단계로 진행하여 테스트를 계속 진행하고, 없으면 테스트를 종료하는 단계로 이루어진다.

<20> 이하, 본 발명에 따른 소프트웨어 테스트 시스템 및 방법을 첨부한 도면들을 참조하여 다음과 같이 설명한다.

<21> 도 2는 본 발명에 따른 소프트웨어 테스트 시스템의 개념을 나타내는 도면이다. 본

발명에 따른 소프트웨어 테스트 시스템은 함수 라이브러리 파일(10), 오브젝트 파일(14) 및 실행 프로그램(12)으로 이루어지며, 설명의 편의를 위해 도 2에는 테스트하고자 하는 타깃 소프트웨어(16)를 함께 도시한다. 여기서, 타깃 소프트웨어(16) 및 실행 프로그램(12)은 컴퓨터에서 실행 가능한 프로그램 코드로 이루어진다. 또한, 실행 프로그램(12)은 레이셔널 팀테스트가 제공하는 프로그램 언어 SQABasic을 이용하여 프로그래밍할 수 있다.

<22> 도 2를 참조하여, 함수 라이브러리 파일(10)에는 타깃 소프트웨어(16)의 오브젝트들을 실행하기 위한 명령어들이 함수로 일반화되어 기록된다. 오브젝트 파일(14)에는 타깃 소프트웨어(16)의 오브젝트를 인식할 수 있는 오브젝트 인식값들이, 타깃 소프트웨어(16)를 테스트하고자 하는 순서에 따라, 순차적으로 기록된다. 실행 프로그램(12)은 오브젝트 파일(14)로부터 오브젝트 인식값을 순차적으로 읽어들이어 실행하고자 하는 오브젝트를 인식하고, 인식된 오브젝트를 실행하기 위한 함수를 함수 라이브러리 파일로부터 호출하여 함수를 실행한다. 여기서, 함수의 실행은 컴퓨터상에서 타깃 소프트웨어(16)의 테스트 실행을 의미하며, 사용자는 컴퓨터를 통해 테스트 실행 상태를 관찰하며 테스트 실행 결과를 얻을 수 있다.

<23> 도 3a 내지 도 3c는 도 2에 도시된 오브젝트 파일(14), 실행 프로그램(12) 및 함수 라이브러리 파일(10)의 일 예를 각각 나타내는 도면이다. 도 3a는 오브젝트 파일(14)의 일 예로서, 타깃 소프트웨어의 오브젝트를 인식할 수 있는 오브젝트인식값들 '메뉴선택', '파일(F)→열기(O)', '버튼', '열기', '취소'등이 이 테스트 순서에 따라 순차적으로 기록되며, 각 오브젝트 인식값들은 오브젝트 구분자인 쉼표(,)로 각각 구분된다.

<24> 도 3b는 실행 프로그램(12)의 일 예로서, 도 3a에 도시된 오브젝트 파일로부터 오

브젝트 인식값 '메뉴선택'을 읽어들이 '메뉴선택'에 해당되는 함수가 'Menu'임을 인식하고, 함수 라이브러리 파일로부터 'Menu' 함수를 호출하여 실행한다.

<25> 도 3c는 함수 라이브러리 파일의 일 예로서, 명령어 'Window SetContext'와 'PushButton Click'이 함수 'Setf(b)'와 'CancelBut(a)'로서 각각 일반화됨을 보인다.

<26> 도 4는 본 발명에 따른 소프트웨어 테스트 방법의 일 실시예를 설명하기 위한 플로우차트이다.

<27> 도 2 내지 도 4를 참조하여, 먼저 테스트하고자 하는 타깃 소프트웨어(16)를 테스트하기에 앞서, 타깃 소프트웨어(16)의 오브젝트들을 실행하기 위한 다수의 명령어들을 함수로 일반화시켜 도 3c에 도시된 바와 같은 함수 라이브러리 파일(10)을 생성한다(제20단계). 제20단계 후에, 오브젝트 인식값으로부터 타깃 소프트웨어(16)의 오브젝트를 인식하고, 인식된 오브젝트를 실행하기 위한 함수를 함수 라이브러리 파일(10)로부터 호출하며, 호출된 함수를 실행하여 타깃 소프트웨어의 테스트를 수행하는 도 3b에 도시된 바와 같은 실행 프로그램(12)을 생성한다(제22단계). 제22단계 후에, 타깃 소프트웨어의 오브젝트를 나타내는 오브젝트 인식값을 타깃 소프트웨어를 테스트하고자 하는 테스트 순서에 따라 순차적으로 기록한 도 3a에 도시된 바와 같은 오브젝트 파일(14)을 생성한다(제24단계).

<28> 제24단계 후에, 실행 프로그램(12)은 오브젝트 파일(14)로부터 오브젝트 인식값을 하나씩 순차적으로 인식한다(제26단계). 이 때, 오브젝트 파일에 기록된 오브젝트들은 도 3a에 도시된 바와 같이 오브젝트 구분자(,)에 의해 각각 분리되어 인식된다. 예컨대, 실행 프로그램(12)은 도 3a에 도시된 오브젝트 파일(14)로부터 '메뉴 선택'이라는 오브젝

트 인식값을 읽어들인다.

- <29> 제26단계 후에, 실행 프로그램(12)은 읽어들인 오브젝트 인식값에 해당하는 함수가 함수 라이브러리 파일(10)에 존재하는가를 검색한다(제28단계). 예컨대, 제26단계에서 실행 프로그램(12)이 '메뉴 선택'이라는 오브젝트 인식값을 읽어들이며, 실행 프로그램(12)은 '메뉴 선택'을 실행하기 위한 함수 'Menu'가 함수 라이브러리 파일에 정의되어있는가를 검색한다.
- <30> 실행 프로그램(12)은 제26단계에서 읽어들인 오브젝트에 대한 함수가 함수 라이브러리 파일에 정의되어 있으면, 해당 함수를 호출하고(제30단계), 함수 실행에 필요한 함수 인자들로서, 제26단계에서 읽어들인 오브젝트 인식값 다음에 연속되는 오브젝트 인식값을 읽어들이며 함수실행을 한다(제32단계). 예컨대, 실행 프로그램은 오브젝트 파일에서 '메뉴선택'에 연속되는 오브젝트 인식값 '파일(F)→열기(O)'를 'Menu'함수를 실행하기 위한 인자로서 읽어들이며 'Menu'함수를 실행한다. 여기서, 함수의 실행은 컴퓨터상에서 타깃 소프트웨어의 테스트의 실행을 의미하며, 컴퓨터상에서 그 실행결과를 확인할 수 있다.
- <31> 한편, 제28단계에서, 제26단계에서 읽어들인 오브젝트 인식값에 해당하는 함수가 함수 라이브러리 파일(10)에 정의되어있지 않다면 테스트를 종료한다.
- <32> 제32단계 후에, 실행 프로그램은 오브젝트 파일(14)에 실행되지 않은 오브젝트 인식값들이 남아있는가를 판단하여, 오브젝트 인식값들이 남아있으면 테스트를 계속하기 위해 다음 오브젝트 인식값을 읽어들이며, 실행되지 않은 오브젝트 인식값이 없다면 테스트가 완료된 것으로 판단하고 테스트를 종료한다(제34단계).
- <33> 이상에서와 같이, 본 발명에 따른 소프트웨어 테스트 시스템 및 방법에 따르면, 관

리대상이 되는 스크립트 파일이 없으므로, 타깃 소프트웨어의 변경에 따라 스크립트 파일을 고치는데 따르는 유지 및 보수 시간이 없다. 즉, 타깃 소프트웨어의 변경이 발생하여도 오브젝트 인식값이 변경되지 않았다면 오브젝트 파일을 변경할 필요없이 그대로 사용하면 된다. 따라서, 타깃 소프트웨어가 변경되어도 함수 라이브러리 파일, 실행 프로그램 또는 오브젝트 파일을 변경할 필요가 없으므로, 타깃 소프트웨어의 변경에 따른 유지 및 보수 공수가 발생되지 않는다.

<34> 또한, 타깃 소프트웨어에 중대한 변경이 발생하여 오브젝트 인식값이 변경되더라도 오브젝트 인식값을 기록한 오브젝트 파일을 고치는데 드는 공수만이 필요하며, 오브젝트 파일을 수정하는 데 드는 공수는 0.5시간 이내이다.

<35> 결국, 타깃 소프트웨어의 변경에 맞추어 테스트 시스템의 유지 및 보수작업은 0.5시간 이내에 이루어질 수 있다. 종래에는 스크립트수, 소프트웨어의 변경율, 프로젝트 규모등 많은 변수가 소프트웨어 테스트 시스템의 유지 및 보수에 드는 공수에 영향을 미친다. 그러나, 본 발명에 따른 소프트웨어 테스트 시스템은 타깃 소프트웨어에 중대한 설계 변경이 발생한 경우에만 유지 및 보수 공수가 들며, 이 때에도 오브젝트 파일을 수정하는 정도로 유지 및 보수는 매우 간단히 이루어진다. 또한, 본 발명에 따른 소프트웨어 테스트 시스템 및 방법을 다른 소프트웨어 및 제품 테스트에 적용시 함수 라이브러리 파일에서 일부 함수를 재정의하는 것만으로 쉽게 적용할 수 있다.

<36> 본 발명은 또한 컴퓨터로 읽을 수 있는 기록매체에 컴퓨터가 읽을 수 있는 코드로서 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록장치를 포함한다. 컴퓨터가 읽을 수 있는 기록매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플라피디스크, 광데이터

저장장치 등이 있으며, 또한 캐리어 웨이브(예를 들어 인터넷을 통한 전송)의 형태로 구현되는 것도 포함한다. 또한 컴퓨터가 읽을 수 있는 기록매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어, 분산방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다.

【발명의 효과】

<37> 상술한 바와 같이, 본 발명에 따른 소프트웨어 테스트 시스템 및 방법은 타겟 소프트웨어의 변경에 따른 유지 및 보수가 매우 간단히 이루어지며, 다른 소프트웨어 및 제품 테스트에 적용시 함수 라이브러리 파일에서 일부 함수를 재정의하는 것만으로 쉽게 적용할 수 있다.

【특허청구범위】**【청구항 1】**

컴퓨터상에서 실행되는 소프트웨어를 테스트하는 테스트 시스템에 있어서,

상기 소프트웨어의 오브젝트들을 실행하기 위한 명령어들을 함수화하여 기록한 함수 라이브러리 파일;

상기 소프트웨어의 각 오브젝트를 나타내는 오브젝트 인식값을 상기 소프트웨어를 테스트하고자 하는 순서에 따라 순차적으로 기록하며, 각 오브젝트 인식값들은 오브젝트 구분자에 의해 구분되는 오브젝트 파일; 및

상기 오브젝트 파일로부터 오브젝트 인식값을 순차적으로 읽어들이어 실행하고자 하는 오브젝트를 인식하고, 인식된 오브젝트를 실행하기 위한 함수를 함수 라이브러리 파일로부터 호출하여 실행하는 실행 프로그램을 포함하는 것을 특징으로 하는 소프트웨어 테스트 시스템.

【청구항 2】

컴퓨터상에서 실행되며, 테스트하고자 하는 타겟 소프트웨어의 명령어를 함수로 일반화시킨 함수 라이브러리 파일을 구비하는 소프트웨어 테스트 시스템에서 상기 타겟 소프트웨어를 테스트하는 방법에 있어서,

(a) 상기 타겟 소프트웨어의 각 오브젝트를 나타내는 오브젝트 인식값을 상기 소프트웨어를 테스트하고자 하는 순서에 따라 순차적으로 기록하며, 각 오브젝트 인식값들은 오브젝트 구분자에 의해 구분되는 오브젝트 파일을 생성하는 단계;

(b)상기 오브젝트 파일에 기록된 오브젝트 인식값들을 하나씩 순차적으로 읽어들이

고, 읽어들인 오브젝트 인식값에 해당하는 오브젝트를 실행하기 위한 함수를 상기 함수 라이브러리 파일로부터 호출하는 단계;

(c) 상기 (b)단계에서 호출된 함수의 실행에 필요한 소정 개수의 함수 인자들로서 상기 (a)단계에서 읽어들인 오브젝트 인식값 다음에 연속하는 오브젝트 인식값들을 읽어 들여, 상기 (b)단계에서 호출된 함수를 실행하는 단계; 및

(d)상기 오브젝트 파일에 실행되지 않은 오브젝트 인식값이 있으면 상기 (b)단계로 진행하여 테스트를 계속 진행하고, 없으면 테스트를 종료하는 단계로 이루어지는 것을 특징으로 하는 소프트웨어 테스트 방법.

【청구항 3】

컴퓨터상에서 실행되며, 테스트하고자 하는 타겟 소프트웨어의 명령어를 함수로 일반화시킨 함수 라이브러리 파일 및 상기 타겟 소프트웨어의 각 오브젝트를 나타내는 오브젝트 인식값을 상기 소프트웨어를 테스트하고자 하는 순서에 따라 순차적으로 기록하며, 각 오브젝트 인식값들은 오브젝트 구분자에 의해 구분되는 오브젝트 파일을 구비하는 소프트웨어 테스트 시스템에서 상기 타겟 소프트웨어를 테스트하는 프로그램을 기록한 기록 매체에 있어서,

(a) 상기 오브젝트 파일에 기록된 오브젝트 인식값들을 하나씩 순차적으로 읽어들이고, 읽어들인 오브젝트 인식값에 해당하는 오브젝트를 실행하기 위한 함수를 상기 함수 라이브러리 파일로부터 호출하는 단계;

(b)상기 (a)단계에서 호출된 함수의 실행에 필요한 소정 개수의 함수 인자들로서 상기 (a)단계에서 읽어들인 오브젝트 인식값 다음에 연속하는 오브젝트 인식값들을 읽어

들여, 상기 (a)단계에서 호출된 함수를 실행하는 단계; 및

(c)상기 오브젝트 파일에 실행되지 않은 오브젝트 인식값이 있으면 상기 (a)단계로 진행하여 테스트를 계속 진행하고, 없으면 테스트를 종료하는 단계로 이루어지는 것을 특징으로 하는 소프트웨어 테스트 프로그램을 컴퓨터에서 실행 가능한 프로그램 코드로 기록한 기록 매체.

【도면】

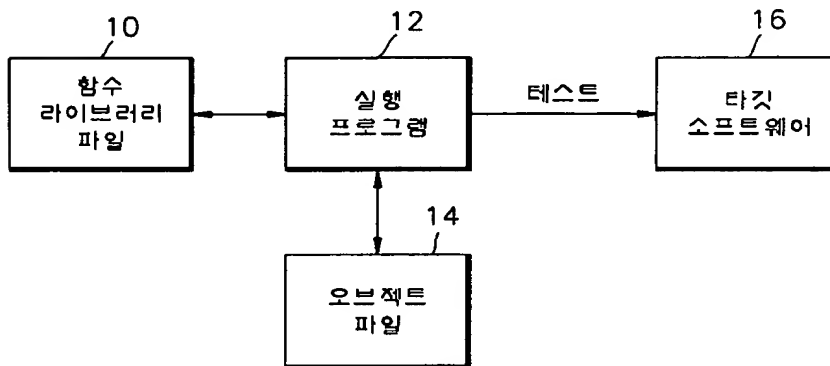
【도 1】

```

1 Window SetContext, "Caption=제목 없음 -메모장", ""
  MenuSelect "파일(F)->열기(O)..."
  ...
n Window SetContext, "Caption=열기", ""
  PushButton Click, "Text=취소"

```

【도 2】



【도 3a】

메뉴 선택, 파일(F) —> 열기(O)...., 버튼, 열기, 취소,.....

【도 3b】

```

.....
do while MyArray(i) <> ""
  if MyArray(i) = "메뉴선택" then
    Menu(MyArray(i + 1))
  else
    end if
.....

```

【도 3c】

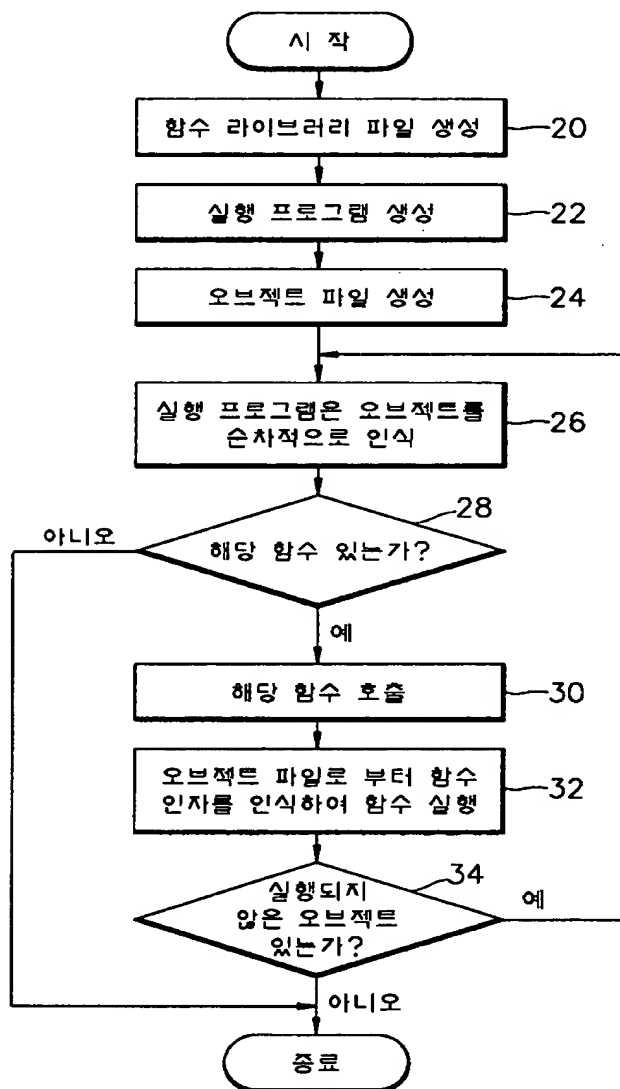
```

Function Setf(b) <----- 함수시작
Window SetContext, "Caption="+b+"", ""
End Function <----- 함수 끝

Function CancelBut(a)
PushButton Click, "Text="+a+"
End Function

```

【도 4】



J1046 U.S. PTO
09/851405



KOREAN INDUSTRIAL PROPERTY OFFICE

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Industrial Property Office.

Application Number: Patent Application No. 00-51651

Date of Application: 1 September 2000

Applicant(s): Samsung SDS Co., Ltd.

5 February 2001

COMMISSIONER

1020000051651

2001/2/5

[Document] Application
[Right] Patent
[Receiver] Commissioner
[Filing Date] 1 September 2000
[IPC] H04L
[Title] Software test system and method

[Applicant]
[Name] Samsung SDS Co., Ltd.
[Applicant code] 1-1998-001790-3

[Attorney]
[Name] Young-pil Lee
[Attorney code] 9-1998-000334-6
[General Power of Attorney
Registration No.] 1999-050326-4

[Attorney]
[Name] Heung-soo Choi
[Attorney code] 9-1998-000657-4
[General Power of Attorney
Registration No.] 1999-025556-5

[Attorney]
[Name] Hae-young Lee
[Attorney code] 9-1999-000227-4
[General Power of Attorney
Registration No.] 2000-004535-8

[Inventor]

[Name] Young-seok Lee
[Resident
Registration No.] 650315-1052030
[Zip code] 136-060
[Address] 101-1408, Samsung Apt., Donam-dong
Seongbuk-gu, Seoul, Rep. of Korea
[Nationality] Rep. of Korea
[Priority Claiming]
Country: Republic of Korea
Classification: Patent

1020000051651

2001/2/5

Application Number: 10-2000-0024608

Filing Date: 9 March 2000

Documentary evidence Attached

[Request for Examination] Yes

[Purpose] We file the present application according to Article 42 of the Patent Law
also Request for Examination according to Article 60 of the Patent Law.

Attorney

Young-pil Lee

Attorney

Heung-soo Choi

Attorney

Hae-young Lee

[Fee]

[Basic fee]	20 Sheet(s)	29,000 won
-------------	-------------	------------

[Additional fee]	23 Sheet(s)	23,000 won
------------------	-------------	------------

[Priority claiming fee]	1 Case(s)	26,000 won
-------------------------	-----------	------------

[Examination fee]	9 Claim(s)	397,000 won
-------------------	-------------	-------------

[Total]		475,000 won
---------	--	-------------

[Enclosures]

1. Abstract and Specification (and Drawings) 1 original